Z8 BASIC System/Controller Board

USER MANUAL AND ASSEMBLY GUIDE

THE MICROMINT, INC. 561 Willow Avenue, Cedarhurst, N.Y. 11516

THE MICROMINT INC. Z8 BASIC SYSTEM/CONTROLLER BOARD

- Owners' Manual and Assembly Instructions

Copyright 1983 The MICROMINT Inc.

TABLE OF CONTENTS

		Page
I	Introduction	4
II	Microprocessors Vs Single Chip Microcomputers	6
III	The ZILOG Z8671 Microcomputer	8
IV	Z8 On-board Memory Configurations	10
V	The Z8 BASIC SYSTEM/CONTROLLER Board	11
VI	The BASIC/DEBUG Monitor	14
VII	Memory Allocation	15
VIII	Program Format and Storage	17
IX	Parallel I/O Expansion	20
XI	Memory Mapped I/O Port and Baud Rate Switches	21
XI	Z8 System Configuration Jumpers	22
XII	Assembling the Z8 BASIC SYSTEM/CONTROLLER	24
XIII	Z8 Universal Power Supply	27
XIV	Z8 Standard Power Supply	28
XV	Warranty	29

TABLE OF FIGURES

		Page
I	Technical Specifications	5
II	Z8 Architecture	6
III	Z8671 Pinout	7
IV	Z8671 Memory Allocation	7
V	Z8671 Internal Memory Allocation	9
VI	Z8 BASIC SYSTEM/CONTROLLER Silkscreen	11
VII	Z8 Schematics	12
VIII	Z8 BASIC SYSTEM/CONTROLLER Parts Lists	13
IX	Z8 BASIC SYSTEM/CONTROLLER Memory Map	16
X	Z8 Hex Dump Program	19
XI	Z8 Expansion Connector J1	26
XII	Universal Power Supply Schematic	27
XIII	Universal Power Supply Parts List	27
XIV	Standard Power Supply Schematic	28
XV	Standard Power Supply Parts List	28

INTRODUCTION

The MICROMINT INC Z8 BASIC SYSTEM/CONTROLLER board is a milestone in microcomputer price-performance. It is cheap enough to dedicate to a specific application, intelligent enough to be programmed directly in a high level language, and efficient enough to be battery operated if required. Depending upon your needs, it can be utilized as an inexpensive tiny BASIC computer for general interest or dedicated to specialized tasks such as security control, energy management, solar system monitoring, or intelligent peripheral control.

The entire computer is 4" by 4 1/2" and includes a tiny BASIC interpreter, up to 6K bytes of program memory in ROM, or up to 4K bytes of RAM, one RS-232 serial port and two parallel ports, plus a variety of other features (A condensed functional specification is outlined in figure 1). Using a Z8 microcomputer chip and 2016, 2716 or 2732 memory chips, the Z8 BASIC SYSTEM/CONTROLLER board is completely self contained and optimized for use as a dedicated controller. For example, to program it for a dedicated application you would merely attach a terminal to the RS-232 connector, power up the system, and type in a BASIC program using familiar commands such as GOTO, IF, GOSUB, LET, etc. Executing the program is simply a matter of typing RUN. For higher speed applications or just to experiment with the Z8 instruction set, a USR command facilitates calls to machine language subroutines.

Once a program has been written and tested with the aid of the terminal, the finished program can be transferred to an EPROM via a memory dump program and the terminal removed. Next, the 2K RAMS are removed from their sockets and either two 2716 (2K X 8) or one 2732 (4K X 8) EPROM (depending upon the length of the program) plugged into the 24 pin socket(s). THE EPROMS AND 2016 RAMS ARE PIN COMPATIBLE. PERMANENT PROGRAM STORAGE IS SIMPLY A MATTER OF PLUGGING EPROM(S) INTO THE RAM SOCKET(S). When the Z8 board is powered up, the stored program is immediately executed. More on all these features later.

Z8 BASIC MICROCOMPUTER BOARD (Figure 1)

Technical Specifications:

Processor	Zilog Z8 8-bit microcomputer with RAM, ROM and I/O in a single package. Z8671 includes a 2K tiny BASIC/DEBUG resident interpreter in ROM, 144 bytes RAM, and 32 I/O lines. System runs at 7.3728 MHz clock rate. 1.5 usec instruction time. 6 interrupts.
Memory	Two on-board sockets allow use of 4K RAM, 6K ROM or 2K RAM and 4KROM. Supports 2016 type RAMs,and single 5V supply 2716 and 2732 EPROMs; 2K BASIC/DEBUG ROM based interpreter; Memory externally expandable to 62K bytes of program memory and 64K bytes of data memory.
Input/Output	Serial Port --- RS-232C compatible and switch selectable to 110, 150, 300, 1200 2400, 4800, 9600 bps. Parallel I/O --- 2 parallel ports: one dedicated input and one bit-programmable as input or output. Programmable interrupt and handshaking lines; All signals are LSTTL compatible.
External I/O	16 bit address and 8 bit bidirectional data bus brought out to expansion connector.
BASIC Commands	GOTO, GO@, (USR), GOSUB, IF THEN, INPUT, LET, LIST, NEW, REM, RETURN, RUN , STOP, IN, PRINT, PRINT HEX. Integer arithmetic: +,-,/,*,AND BASIC can call machine language sub-routines for increased execution speed; Allows complete memory and register interrogation and modification.
Power Supply ..... Requirements	+5 Volts +/- 5% @ 250 mA. +12 Volts +/-30% @ 30 mA. -12 Volts +/-30% @ 30 mA. (The 12 volt supplies are only required for RS-232C operation.)
Dimensions and ..... Connections	4" by 4 1/2" board, dual 22 pin (.156") edge connector. 25 pin D-subminiature RS-232C receptacle , 8 pole DIP switch data rate selector or I/O port accessed by 9 pin BERG strip.
Operating ..... Conditions	Temperature: 0-50C (32-122F) Relative Humidity: 10-90% non condensing

MICROPROCESSORS VERSUS SINGLE CHIP MICROCOMPUTERS

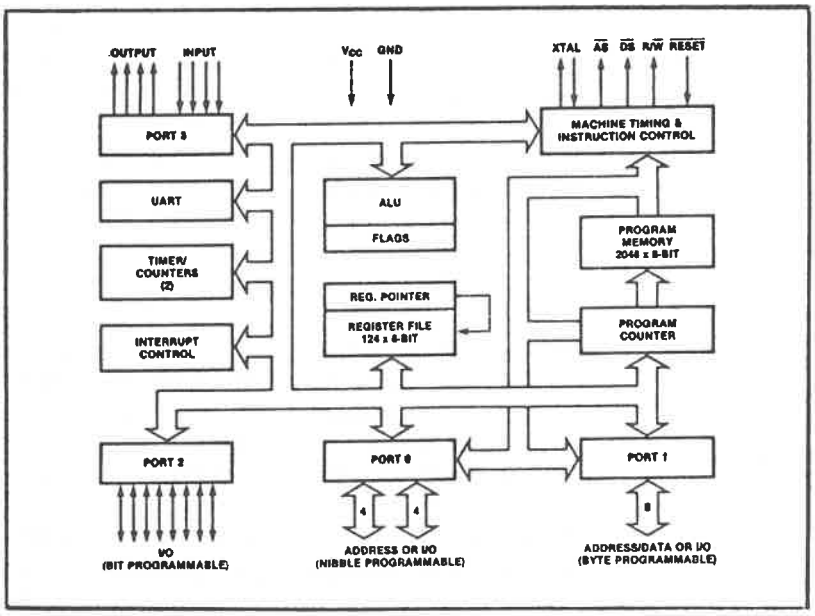
The central component in this miniature computer is the ZILOG Z8671 (while we have been simply calling it the Z8, this is a generic term referring to a family of devices. The Z8671 is just one of them). Rather than a microprocessor like the Z80, the Z8 is a single chip microcomputer. It contains RAM, ROM, and I/O as well as other standard CPU functions. Microprocessors such as the Z80 or 8080 require support circuitry to build a functional computer. A single chip micro-computer, on the other hand, can function solely on its own with no other chips.

The concept is not new. Single chip microcomputers have been around for quite a while and millions of them are used in electronic games. The Z8, however, raises the capabilities of single chip computers to new heights and provides many powerful features usually found only in general application micro-processors.

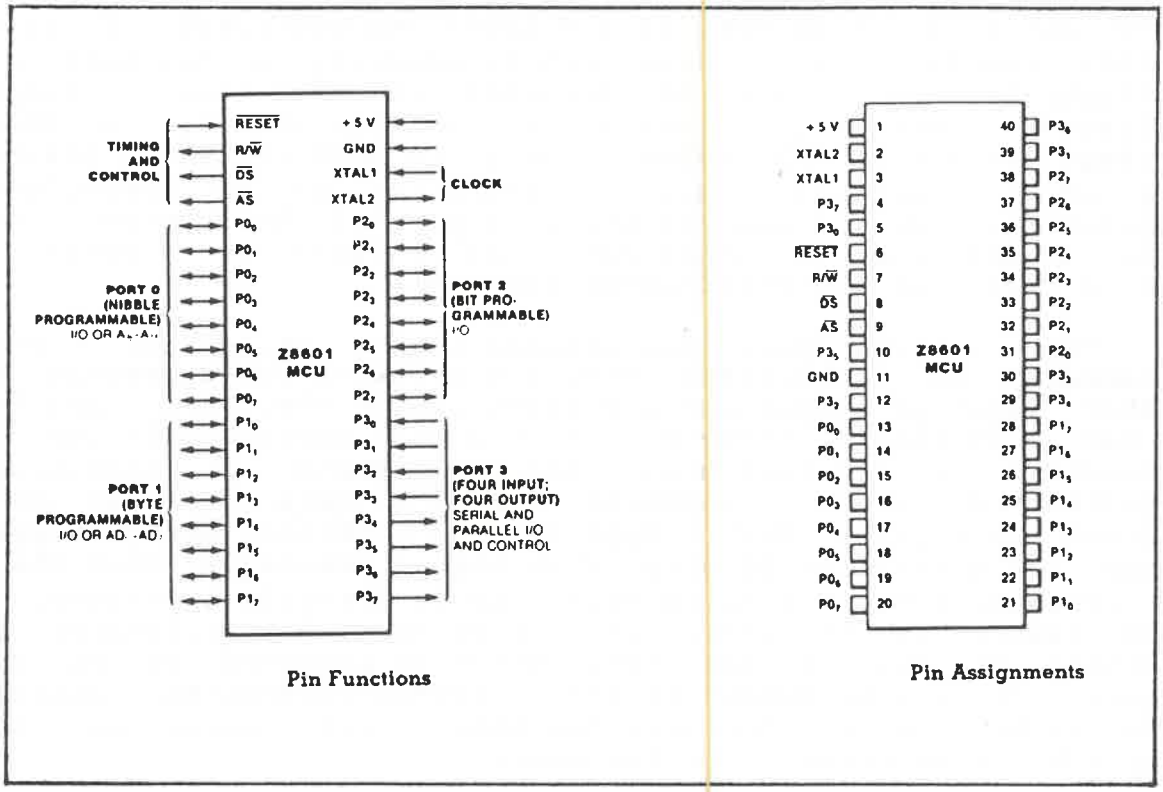
Traditionally, single chip microcomputers have been designed for microcontroller applications and optimized for I/O manipulation. On a 40 pin package, often as many as 32 of the pins can be I/O related. A ROM programmed single chip microcomputer used in an electronic chess game might offer a thousand variations of game strategy but it could not be reprogrammed to function as a word processor. The ability to reorient processing functions and reallocate memory has generally been the province of memory intensive microprocessors.

The Z8 architecture (shown in figure 2) allows it to serve in either memory or I/O intensive applications and function to a great extent as a microprocessor. Under program control the Z8671 can be configured as a stand alone microcomputer with 2K of internal ROM, as a traditional microprocessor with 124K of external memory, or as a parallel processing element with other computers. Conceivably, the Z8 might just as easily be used as a single chip controller in a microwave oven as the CPU in a point-of-sale terminal complete with floppy disks.

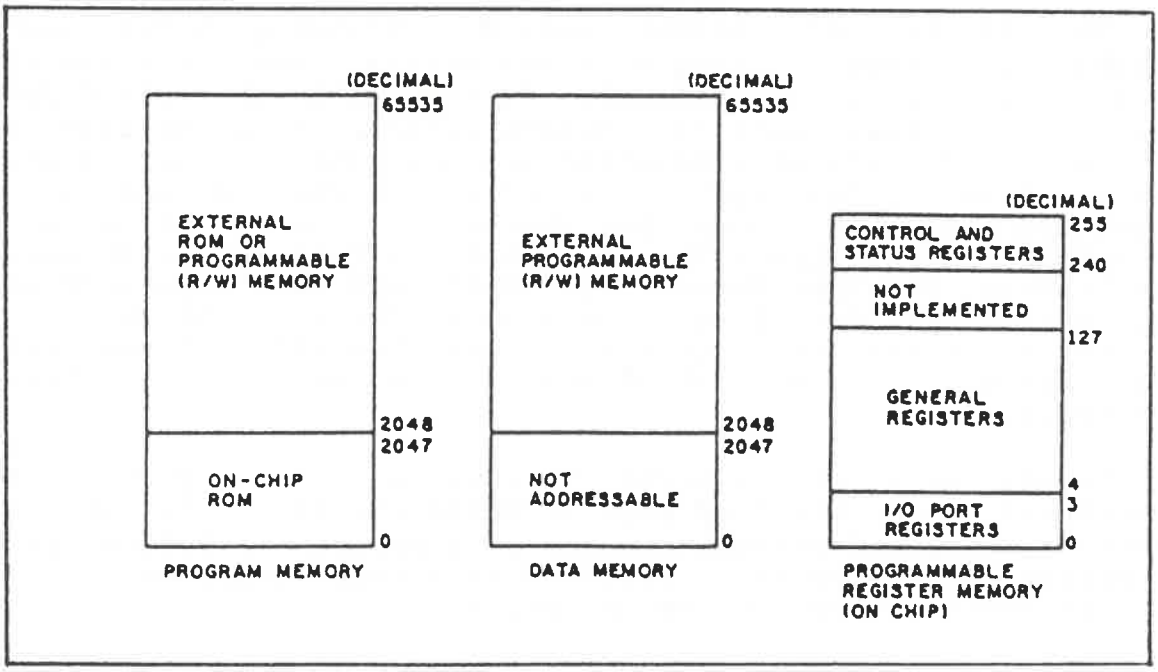
Z8 ARCHITECTURE (Figure 2)



Z8671 PINOUT (Figure 3)



Z8671 MEMORY ALLOCATION (Figure 4)



The Z8671 contains 2K bytes of ROM (preprogrammed with a tiny BASIC interpreter), 32 I/O lines and 144 bytes of random access memory. The internal RAM is actually a register file (illustrated in figure 5) composed of 124 general purpose registers (R4-R127), 1 serial I/O register (R240), 15 status control registers (R241-R255), and 4 I/O port registers (R0-R3). Any of the general purpose registers can be an accumulator, address pointer, index register, or part of the internal stack (the significance of these registers is explained in detail in the Z8 Technical and BASIC/DEBUG manuals).

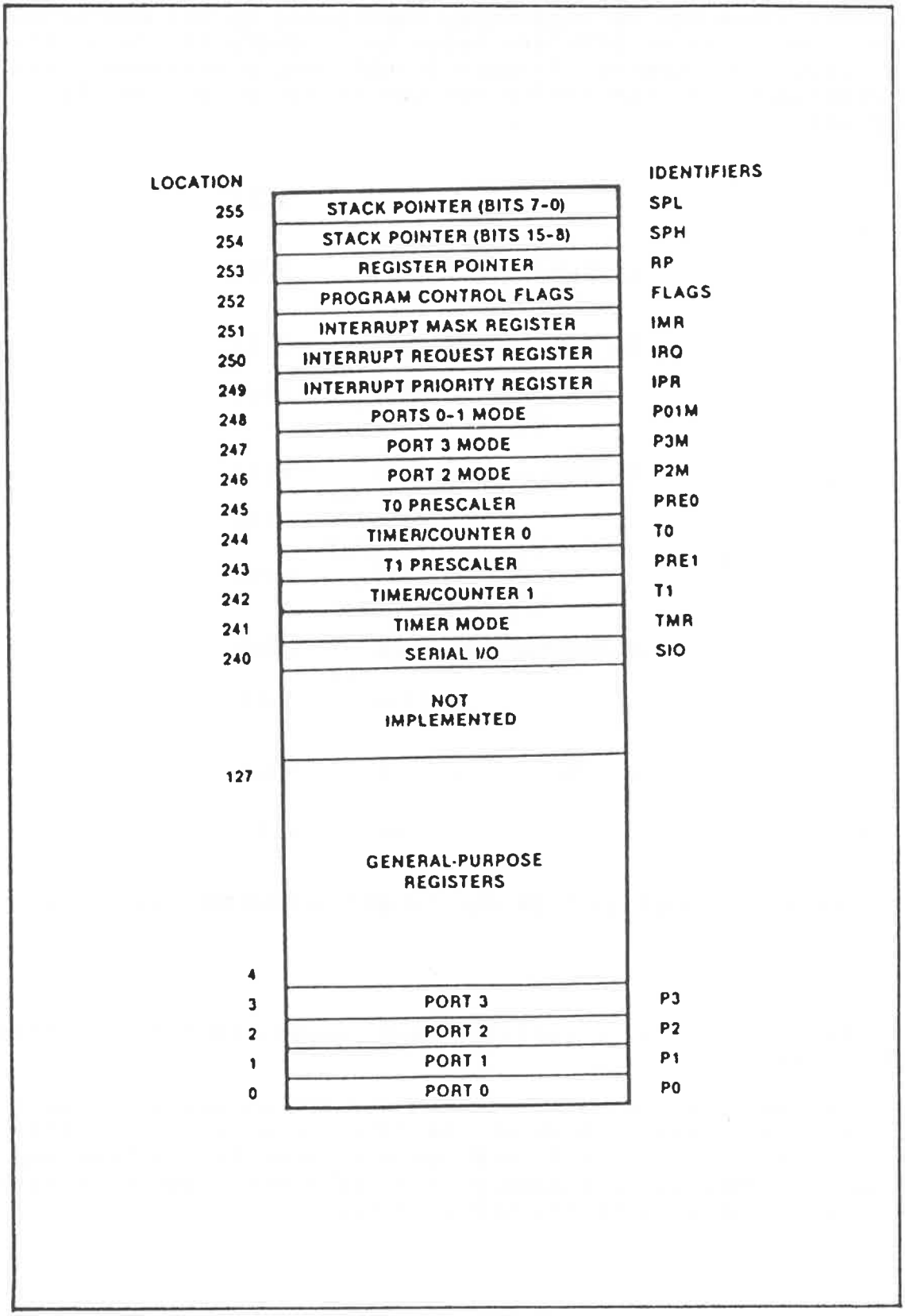
The 32 I/O lines are grouped into 4 ports and treated internally as 4 registers. They are software configureable for either input or output and are LSTTL compatible. Port 1 and Port 0 can additionally serve as a multiplexed address/data bus for connection of external memory and peripherals. In traditional nomenclature, Port 1 supplies the D0-D7 data bus and A0-A7 address bus signals. Port 0 supplies the remaining A8-A15 address lines for a total of 16 bits. This allows 62K bytes (plus the 2K of internal ROM) of program memory to be directly addressed. If more memory is required, one bit on Port 3 facilitates the addition of another 64K bytes which is referred to as data memory. In the MICROMINT Z8 BASIC SYSTEM/CONTROLLER, separate data memory is not presently implemented and program and data memory are considered to be the same.

The address and data buses are brought out to a 44 pin expansion connector. These signals are fully LSTTL compatible as mentioned. However, if more than 2 additional LSTTL gates are connected to the address and data strobe lines, additional buffering may be necessary.

The Z8 has 47 instructions, 9 addressing modes and 6 maskable interrupts. Using an 8 MHz crystal most instructions average 1.5 to 2.5 microseconds. In the Z8671 the BASIC/DEBUG interpreter allows machine language programs to be written and then executed through subroutine calls in BASIC. This feature greatly enhances the capabilities of a tiny computer and allows connection to high speed peripherals. The final area of concern is communication. The Z8 contains a full duplex Universal Asynchronous Receiver/Transmitter (UART) and two counter/timers with prescalers. One of the counters divides a 7.3728 MHz clock to one of 8 standard data rates. With the Z8671 these rates range between 110 and 19200 bps and are switch or software selectable.

Serial data is received through bit 0 of Port 3 and transmitted from bit 7 of Port 3. While the Z8 can be set for either even or odd parity, the Z8671 is preset for 8 data bits, no parity and 2 stop bits. Received data must always have 8 data bits, at least 1 stop bit, and no parity.

Z8671 INTERNAL RAM MEMORY MAP (Figure 5)



Z8 ON-BOARD MEMORY CONFIGURATIONS.

The Z8 BASIC SYSTEM/CONTROLLER has two 24 pin memory sockets. These can be separatley configured to use ROM or RAM. 2K RAMs, and 2 or 4K ROMs are supported, namely the 2016 RAM, and the 2716/2732 EPROMs. (Single 5 volt supply versions only.) 5 combinations of RAM and/or ROM can be set up on the Z8 board. They are :

	IC10	IC11
4K RAM.....	2016	2016
2K RAM/2K ROM.	2016	2716
	or	
	2716	2016
4K ROM.....	2716	2716
	or	
	none	2732
	or	
	2732	none
2K RAM/4K ROM.	2016	2732
	or	
	2732	2016
6K ROM.....	2716	2732
	or	
	2732	2716

>>>>>>>>>>NOTE THAT 8K ROM IS NOT SUPPORTED<<<<<<<<<<

Use of the memory configuration jumpers is covered later in this manual.

The ability to have both RAM and ROM on the main board at the same time greatly enhances the Z8's use as a data acquisition and control system. The 6K ROM support capability allows the use of larger BASIC or Z8 assembly language control programs without the need to use an external memory card.

THE Z8 BASIC SYSTEM/CONTROLLER BOARD

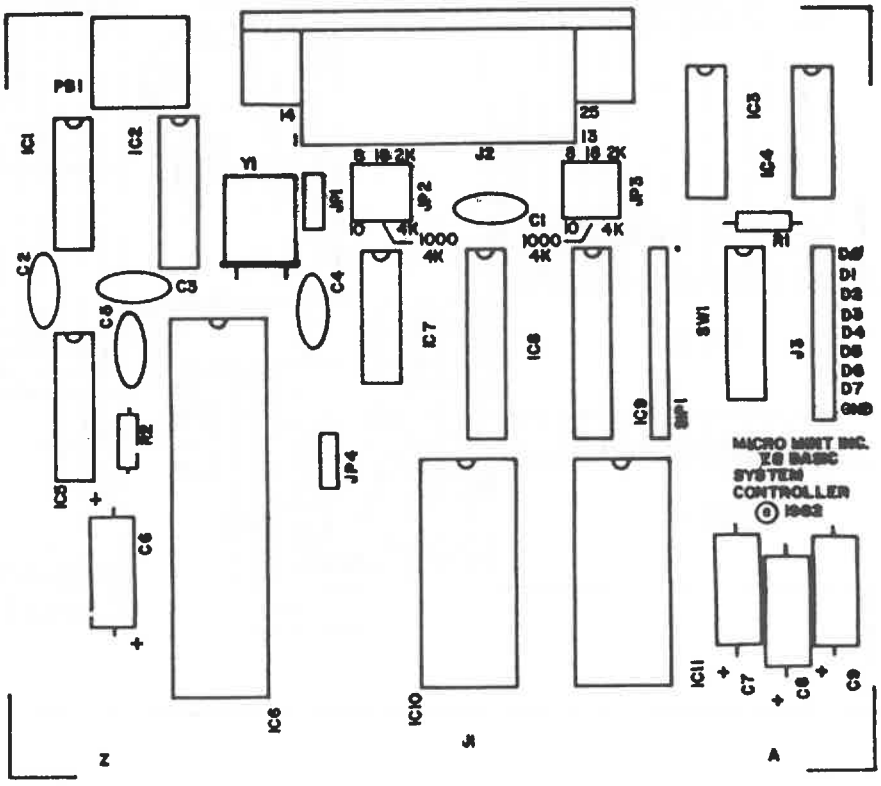
Figure 7 outlines the circuit of the 11 chip Z8 BASIC SYSTEM/CONTROLLER board (Figure 8 is a complete parts list). IC 6 is the Z8671 microcomputer containing Zilog's 2K BASIC/DEBUG ROM monitor software. ICs 10 and 11 are the memory, and IC 8 is an 8 bit address latch. ICs 2 and 9 form a memory mapped input port addressed for either E000-E800 or F800-FFFF hex (it decodes any address within the range selected.) ICs 3 and 4 handle the RS-232 serial communication.

The 11 chip computer typically takes about 225 milliamps at +5 volts. The +12 and -12 volt supplies are only required for RS-232 operation. Power required is typically 25 milliamps on each.

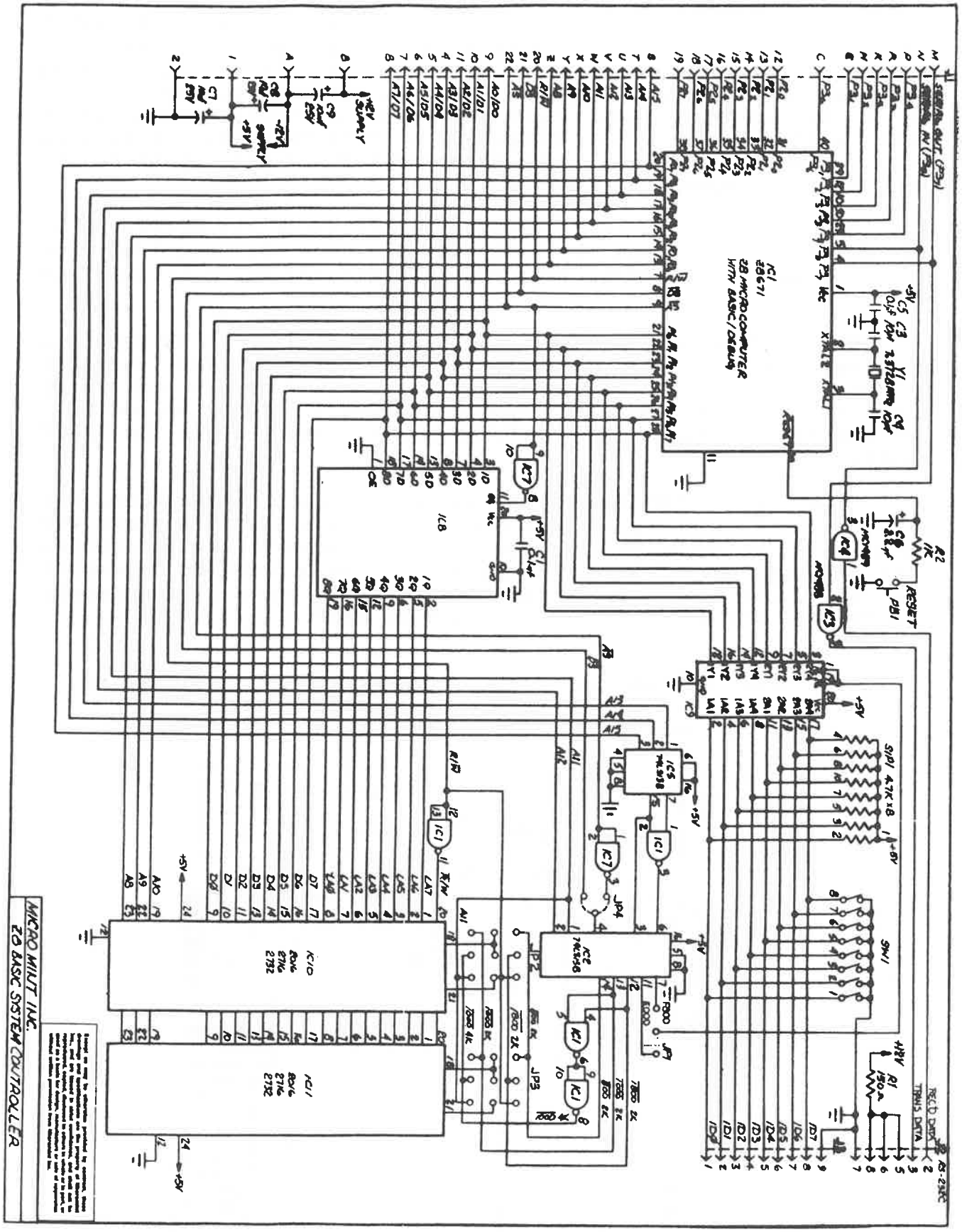
The easiest way to check out the Z8 computer is to attach a terminal to the RS-232 connector and set the data rate selector switch to some convenient rate (8 bit data, no parity and 1 stop bit). For example, to select 1200 baud, SW2 should be on and SW1, SW3 and SW4 are off. After applying power, simply press the RESET pushbutton.

Pressing "RESET" starts the Z8 initialization procedure. One of the things it does is read memory location FFFD hex where the data rate selector switches are located. When it has acquired this information, it sets the appropriate data rate and transmits a colon to the terminal. At this point the Z8 board is completely operational and programs can be typed in BASIC.

THE Z8 BASIC SYSTEM/CONTROLLER SILKSCREEN (Figure 6)



Z8 SCHEMATIC (Figure 7)



MICROMINT Z8 SYSTEM/CONTROLLER PARTS LIST (Figure 8)

Item	Value or Designation
PC Board	Z8 PCB
IC1	74LS00
IC2	74LS138
IC3	MC1488
IC4	MC1489
IC5	74LS138
IC6	Z8671 - BASIC/DEBUG
IC7	74LS00
IC8	74LS373
IC9	74LS244
IC10	2016, 2716, 2732
IC11	2016, 2716, 2732
J2	AMP 206584-1 or equiv. RS-232 Connector
Y1	7.3728 MHz crystal
SW1	8 pole DIP switch
PB1	Reset Pushbutton
C1,C2,C5	.1MFd. 16VDC
C3,C4	10 PFd. 16VDC
C6	2.2 MFd. 25VDC
C7,C8,C9	10 MFd. 25VDC
SIP1	4.7K SIP
R1	150 Ohms (BRN GRN BRN)
R2	1K (BRN BLK RED)
40 Pin socket	1 piece
24 Pin socket	2 pieces
20 Pin socket	2 pieces
16 Pin socket	2 pieces
14 Pin socket	4 pieces
1 X 3 BERG strip	4 pieces
2 X 3 BERG strip	2 pieces
1 X 9 BERG strip	1 piece

BASIC/DEBUG IS A PROCESS CONTROL ORIENTED BASIC MONITOR

Essentially an integer math subset of Dartmouth BASIC, the Zilog BASIC/DEBUG monitor is specifically designed for process control. BASIC/DEBUG recognizes 15 command words. They include GOTO, GO@, USR, GOSUB, IF/THEN, INPUT, IN, LET, LIST, NEW, REM, RUN, RETURN, STOP, PRINT, and PRINT HEX. Standard syntax and mathematical operators are used.

Twenty six variables (A-Z) are supported. Each letter is a variable name. Variables can be used to designate program line numbers. For example, GOSUB B*100 and GOTO A*B*C are valid expressions. BASIC/DEBUG allows examination and modification of any memory location, I/O port, or register. The interpreter processes data in both decimal and hexadecimal notation and accesses machine language code as either a subroutine or a user defined function.

BASIC/DEBUG can directly address the Z8 internal registers and all external memory. Byte references, which use the "@" character with an address, may be used to modify a single register in the CPU, control I/O device, or memory location. For example, @4096 specifies memory location 4096 and @%F6 specifies the PORT 2 mode control register at decimal location 246. To set 45 in memory location 4096 the command is simply, @4096=45 (or @%1000=%2D).

Command abbreviations are standard with most tiny BASICs but this interpreter allows some extremes if you want to limit program space. For example:

```
IF 1>X THEN GOTO 1000     can be abbr.   IF 1>X 1000

PRINT"THE VALUE IS ";S  can be abbr.  "THE VALUE IS ";S

IF X=Y THEN IF Y=Z THEN PRINT "X=Z"
  can be abbr.
IF X=Y IF Y=Z "X=Z"
```

One important difference between standard Dartmouth BASIC and BASIC/DEBUG is that the latter allows variables to define statement numbers and variable storage is not cleared before a program is run. Commands such as GOSUB X or GOTO A*E-Z are all valid. It is also possible to pass values from one program to another. Generally speaking, the Z8761's BASIC is standard in its approach but features such as those mentioned above serve to extend its capabilities.

The main feature which separates this BASIC from others is the level of documentation supplied with the Z8671. While the 2K source code for BASIC/DEBUG is not available (since it is in ROM it can't be changed anyway), the location of all variables, pointers, stacks, etc., are fixed and their locations are defined and described in detail in the BASIC/DEBUG manual.

The 2K interpreter is extremely powerful. Because it operates so easily on register and memory locations, arrays and blocks of data can be easily manipulated. The Z8 BASIC SYSTEM/CONTROLLER with it's interpretive language, virtually eliminates the need for costly development systems, memory consuming editors, assemblers, and debug programs.

MEMORY ALLOCATION

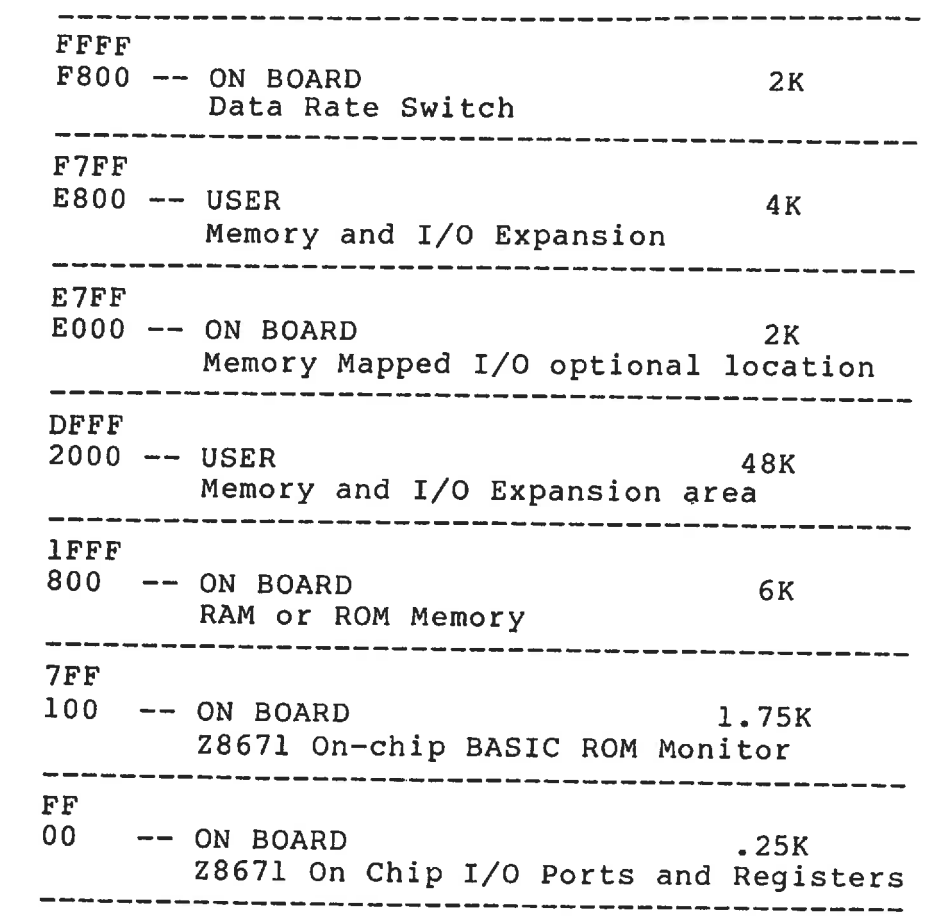
The Z8 has three kinds of memory: registers, internal ROM, and external ROM or RAM. BASIC/DEBUG assigns address 0 to 255 to the register file (no registers are implemented between 128-239 inclusive). The 144 registers include 4 I/O port registers, 124 general purpose registers, 1 serial I/O port, and 15 status and control registers. The 2K byte ROM on the Z8 chip contains the BASIC/DEBUG interpreter. It extends between address 0 through 2047. User memory may start at address 2048 (800 hex). A memory map of the Z8 BASIC SYSTEM/CONTROLLER board is shown in figure 9.

On power up, BASIC/DEBUG sizes RAM, initializes memory pointers, and checks for an auto start-up program. In a RAM system, the top page of memory is then used for the line buffer, variable storage and GOSUB stack. Program entry will begin at location 800 hex if memory exists there.

When BASIC/DEBUG tests memory and finds no RAM, the internal registers are used to store the variables, line buffer, and GOSUB stack. There are limits on the depth of the stack and the number of variables which can be used simultaneously but it is a relatively significant amount. In a system with no RAM, automatic program execution begins at 1020 hex. A BASIC program with a first line number that is < = 255 must be located starting at 1020H in order for autostart to occur.

The 32 bytes between 1000 and 1020 hex are reserved for jump vectors to user supplied interrupt routines. When none are implemented, FF hex should be inserted in this memory area.

Z8 BASIC SYSTEM/CONTROLLER MEMORY MAP (Figure 9)



Whether or not the program resides in ROM or RAM the format is the same. Each BASIC statement begins with a line number and ends with a delimiter. If you were to connect a terminal to the RS-232 serial port and type in the following line:

```
100 PRINT "TEST"
```

it would be stored in memory beginning at 800 hex as follows:

```
         100      P   R   I   N   T       "   T
800      00  64  50  52  49  4E  54  20  22  54

         E   S   T   "
80A      45  53  54  22  00  FF  FF
```

The first 2 bytes of any statement are the binary equivalent of the line number (100 decimal equals 64 hex). Next is the ASCII statement followed by a delimiter (00) which indicates the end of the line. If it is also the last statement in the program (in this case the only one) the next 2 bytes will be FF FF which designate line number 65535. A multi line program further illustrates this format:

```
100 A=5

200 B=6

3005 "A*B=";A*B
```

```
         100      A   =   5       200      B   =
800      00  64  41  3D  35  00  00  C8  42  3D
         6        3005    "   A   *   B   =   "
80A      36  00  0B  BD  22  41  2A  42  3D  22
         ;   A   *   B
814      3B  41  2A  42  00  FF  FF
```

One final example of this is illustrated in Figure 10. Here is a program written to examine itself. Essentially, it is a memory dump routine which lists the contents of memory in hex. As shown, the 15 line program takes 355 bytes and occupies hex 800 to 963.

There is a reason for explaining the internal program format. One of the major features of this computer is its ability to function with programs residing solely in ROM. Getting the programs into the ROM can be quite another matter however.

The easiest way to write an application program into ROM is to serially transmit the contents of the RAM to an EPROM programmer. (Some of you may only have a manual programmer or one with no communication facility. If you are willing to spend the time you can see how easy it is to dump the RAM contents and manually enter the values into the EPROM programmer).

Normally, the list function or terminal dump routine cannot be used because the listing is filled with extraneous spaces and carriage returns. It is possible however to write a program that transmits the contents of memory with no alterations. The data received by the EPROM programmer would be the exact program to load into the EPROM. An EPROM programmer is also available from MICROMINT that runs on the Z8 system. When used with the Z-8 I/O Expansion board, (required) it allows you to directly save programs on EPROM via a menu driven software driver.

The Z8 has the capability of executing machine language programs directly through the USR and GO@ commands. The serial input and serial output subroutines in the BASIC/DEBUG ROM can be executed independently using these commands. The serial input driver is at location 54 hex and the serial output driver is at location 61 hex. Transmitting a single character is simply: GO@ %61,C where C is the value to be transmitted. Receiving a serial character is simply: C=USR (%54) where C equals the received data.

To dump the entire RAM contents to an external EPROM programmer, the following statements should be included at the end of your program (use a GOTO rather than RUN statement to start program execution):

```
1000 X=%800 :REM BEGINNING OF RAM
1010 GO@ %61,@X :REM TRANSMIT CONTENTS OF LOCATION X
1020 X=X+1 :IF X=%1801 THEN STOP
1030 GOTO 1010
```

Execution begins with a GOTO 1000 and ends when the 4K RAM has been dumped. The transmission rate (110 to 9600 bps) is whatever is selected on the data rate selector switches. Conceivably, this technique can also be used to create a cassette storage capability for the Z8. In theory, a 3 or 4 line BASIC program can be entered in high memory (set the memory pointer (R8 and R9) to put the program there) which reads in serial data and loads it in lower memory. Changing the pointers back to 800 hex allows the newly loaded program to be executed. Since the Z8 already has a serial port, any conventional or specialized FSK modem and tape recorder can be used as the cassette interface. A complete cassette interface with driver software on EPROM is also availible from MICROMINT.

Z8 BASIC HEX DUMP PROGRAM (Figure 10)

```
100 PRINT"ENTER START ADDRESS FOR HEX DUMP ";:INPUT X
102 PRINT"THE LIST IS HOW MANY BYTES LONG ";:INPUT C
103 PRINT:PRINT
105 B=X+8 :A=X+C
107 PRINT"ADDRESS                DATA":PRINT
110 PRINT HEX (X);"      ";
120 GOSUB 300
130 X=X+1
140 IF X=B THEN GOTO 180
150 GOTO 120
180 IF X>=A THEN 250
200 PRINT:PRINT:B=X+8:GOTO 110
250 PRINT : STOP
300 PRINT HEX (@X); :PRINT"    ";
310 RETURN



:RUN


ENTER START ADDRESS FOR HEX DUMP ? 2048
THE LIST IS HOW MANY BYTES LONG ? 5000

ADDRESS                        DATA

800       0    64    50    52    49    4E    54    22
808      45    4E    54    ...........
.
.
.
958       0     1    36    52    45    54    55    52
960      4E     0    FF    FF     0     0     0     0
.
.
```

PARALLEL I/O EXPANSION

The Z8 BASIC SYSTEM/CONTROLLER has two parallel ports in addition to the serial port previously described. One port is a memory mapped input port used to read the baud rate switches (only SW1 thru SW3 are used) and is addressed at FFFD. It is brought out to the 9 pin BERG connector on the Z8 board. Since the baud rate switches are only read once at reset and not needed at all for programs without serial communication, this port is available for general input data.

In the standard configuration Port 2 is available as an 8 bit, bit-programmable I/O port. Depending upon the setting in the Port 2 mode register (R246), it can be all input, all output, or any combination. In addition, certain bits of Port 3 can function as hand shaking lines for Port 2 (see the Z8 Technical Manual, mode control register section). These bits are selected through the Port 3 mode control register (R247). For example, if you wanted to configure Port 2 for output with a data ready strobe, you would set 0 in the Port 2 mode control register and 113 in the Port 3 mode control register. The data ready output and data accepted input lines are on Port 3, bits 1 and 6. These two lines should be jumpered together (pins 39 and 40) to produce the strobe. They are brought out to the expansion connector on pins E and L for external access.

Transmitting a value through Port 2 would be done using the "@" command. Doing this in BASIC is simply:

```
100 @246=0:@247=113 :REM SET PORT 2 TO BE OUTPUT

110 @2=X :REM X EQUALS THE DATA TO BE TRANSMITTED
```

BAUD RATE/DATA PORT SWITCHES

The serial communication rate on the Z8 SYSTEM/CONTROLLER board is selected by reading memory location FFFD on reset. If configured for the F800-FFFF range, this memory location is decoded as an input port with 3 data rate selection switches (on input bits 0, 1, and 2) attached. The switch settings for the various data rates are as follows:

Baud Rate	SW1	SW2	SW3	SW4
150	ON	ON	ON	N/A
19200	OFF	ON	ON	N/A
9600	ON	OFF	ON	N/A
4800	OFF	OFF	ON	N/A
2400	ON	ON	OFF	N/A
1200	OFF	ON	OFF	N/A
110	ON	OFF	OFF	N/A
300	OFF	OFF	OFF	N/A

OFF = Open circuit / logic 1

ON = Closed circuit / logic 0

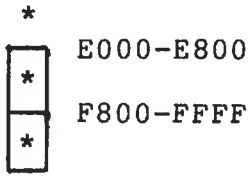
N/A = Don't care

Z8 CONFIGURATION JUMPER

There are 4 jumper pads availible on the Z8 BASIC SYSTEM/CONTROLLER board. These are used to select the addresses and memory type used in the ram/rom sockets, the address range of the memory mapped I/O port, and an OEM option to allow the use of slower EPROMS in an all EPROM system.

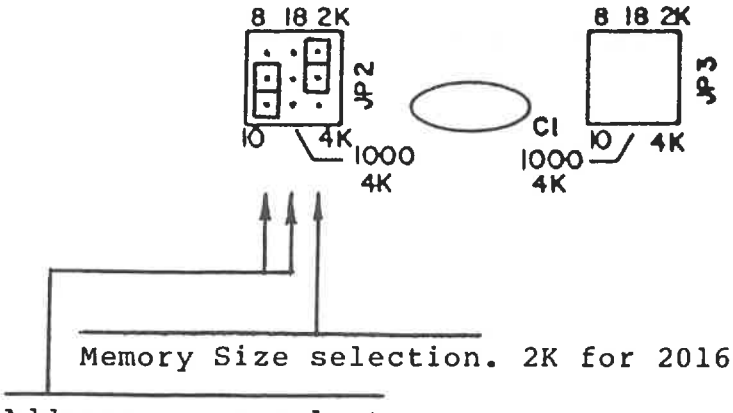
The jumpers are configured as follows :

JP1 - Memory mapped I/O port address range.



The 8 bit memory mapped I/O port will respond to any address in the range selected. It is shown set for the normal configuration, where the I/O port is used to set the baud rate of the RS232 attached terminal.

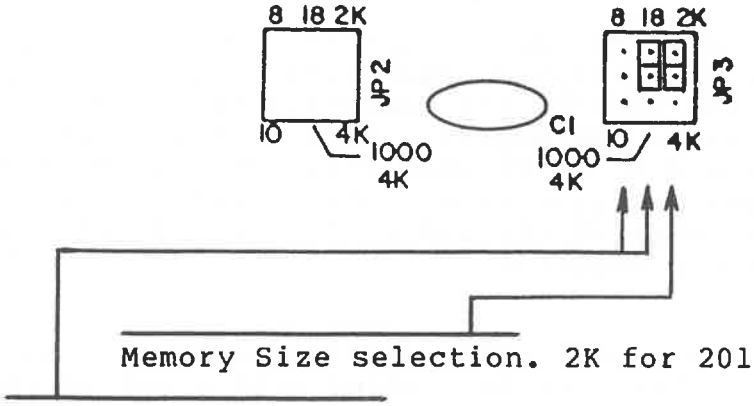
JP2 - IC10 memory socket configuration



This jumper is shown set up for a 2016/2716 addressed at 1000-17FF.

>>>>>>>>>> NOTE <<<<<<<<<<
JP2 and JP3 are interchangeable
as long as the addresses do not conflict

JP3 - IC11 memory socket configuration



Memory Size selection. 2K for 2016/2716. 4K for 2732.

Address range select.
Set a 2K range (800,1000,1800) for 2016/2716.
Set the 4K range (1000) for 2732's

This jumper is shown set up for a 2016/2716 addressed at 1800-1FFF.

JP4 - Extended ROM chip select. (optional)

*
Data Strobe gates logic select
*
Address Strobe gates logic
*

JP4 is normally not used, and is hard wired on the PC board as shown. JP4 allows the use of slow EPROMS in an EPROM ONLY SYSTEM by using the Address Strobe to provide an early chip select line. Using Address Strobe extends the select time by about 250 ns, allowing EPROMS with up to a 650 ns access time to be used. NO RAM MAY BE INSTALLED ON THE Z8 BOARD IF THIS OPTION IS USED !!!!

Remember that the on-board trace must be cut to use the strobe select jumper. USE OF THIS JUMPER VOIDS THE WARRANTY ON PRE-ASSEMBLED BOARDS. THIS MODIFICATION IS INTENDED FOR OEM USE.

ASSEMBLING THE Z8 BASIC SYSTEM/CONTROLLER BOARD

Step 1 --- Insert and solder the IC sockets. Note that there is an indentation or dot designating pin 1. The sockets should line up with the silk screen on the board.

Step 2 --- Insert and solder all resistors and capacitors. Note the polarity of the electrolytic capacitors (C6, C7, C8,and C9) and orient the positive side of the capacitor with the "+" marked on the board. Generally, the negative side of the capacitors are marked with an arrow and a "-" (minus) sign. The opposite side of the capacitor is the positive side. When inserting C2 and C3 carefully bend the leads to fit the 1/4" hole spacing without fracturing the capacitor material.

Step 3 --- Insert and solder SIP1. SIP1 is a 10 pin single-in-line (SIP) package. Pin 1 is designated with a dot on the side of the SIP. It should be inserted such that it lines up with the dot on the silk screen.

Step 4 --- Insert and solder the push button, dip switch and crystal (the crystal should lie flat - a dab of glue helps secure it). Note that the numbers on the dip switch should line up with positions on the silk screen.

Step 5 --- Insert and solder the 25 pin D subminiature connector. With all 25 pins soldered, screws and nuts are not necessary to hold the RS-232 connector.

Step 6 --- Install Jumpers.

NOTE: Unless otherwise specified, jumper positions not designated for connection should be left open. These other jumpers are provided to allow various memory addressing and chip select options on the Z8. Refer to the appropriate manual section for this information.

Step 7 --- Carefully insert ICs 1, 2, 3, 4, 5, 7, 8 and 9 into the sockets. Note that the pin 1 position is designated with either an indentation or dot on the chip.

Step 8 --- Attach the power supply (see note 1 below) and temporarily apply power to the Z8 computer board and measure the three supply voltages. If any of the three voltages (+5, +12, and -12V) is not within operating limits (5% on the 5V, 10% on the + and - 12V), the Z8 board should be inspected for a possible short or improperly inserted IC. Remove the power.

Step 9 -- Insert IC6, IC10 and IC11 into their sockets. Be careful with these devices and treat them as you would a CMOS device (avoid static). Note that the chips have an indentation on one edge. This indentation should line up with the silk screen.

Step 10 --- The easiest way to check out the Z8 computer is to attach a terminal to the RS-232 connector and set the data rate selector switch to some convenient rate (8 bit data, no parity and 1 stop bit). For example, to select 1200 baud, SW2 should be on and SW1, SW3, and SW4 are off. After applying power, simply press the RESET pushbutton. Pressing "RESET" starts the Z8 initialization procedure. One of the things it does is read memory location FFFD hex where the data rate selector switches are located. When it has acquired this information, it sets the appropriate data rate and transmits a colon to the terminal. At this point the Z8 board is completely operational and programs can be entered directly from the terminal in BASIC. You can check the performance of the RAM by entering a multiple line program. The program will not be stored and cannot be listed if the RAM isn't working.

>>>>>>>>>> Note 1 <<<<<<<<<<

The Z8 BASIC SYSTEM/CONTROLLER typically requires 225 milliamps at +5 volts and 25 milliamps each at +12 and -12 volts (the +12V and -12V supplies are only required for RS-232 operation). This power can be supplied through the 44 pin edge connector. Use a 44 pin card edge connector socket (available from The Micromint order part #MB44) and connect ground to pin2, +5v to pin1, +12v to pinB, and -12v to pinA. If you are planning to expand the system, a mother board complete with 5 (44 pin) slots and a 4 pin power supply header is available from The Micromint (order part #MB02).

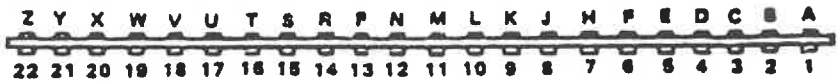
The MICROMINT Inc. manufactures two optionally available triple voltage power supplies.

The Z8 Universal Power Supply, is designed specifically for the Z8 board. It is 2.1" X 4.5" and has a UL listed power plug type transformer. It connects to the mother board through the power plug header. Order part #UPS01 (A&T) or UPS02 (Kit).The UPS01 can provide 300 milliamps at +5V (+/- 5%) and 50 milliamps each at +12V and -12V (+/- 10%).

The Z8 Standard Power Supply, is designed specifically for the Z8 system board which will be expanded into a multiboard system. It is 2.75" X 5" and has a UL listed power plug type transformer. It connects to the mother board through the power plug header. Order part #UPS03 (A&T) or #UPS04 (Kit). The UPS03 can provide 1 amp at +5V (+/- 5%), 300 milliamps at +12V (+/- 5%), and 50 milliamps at -12V (+/- 10%).

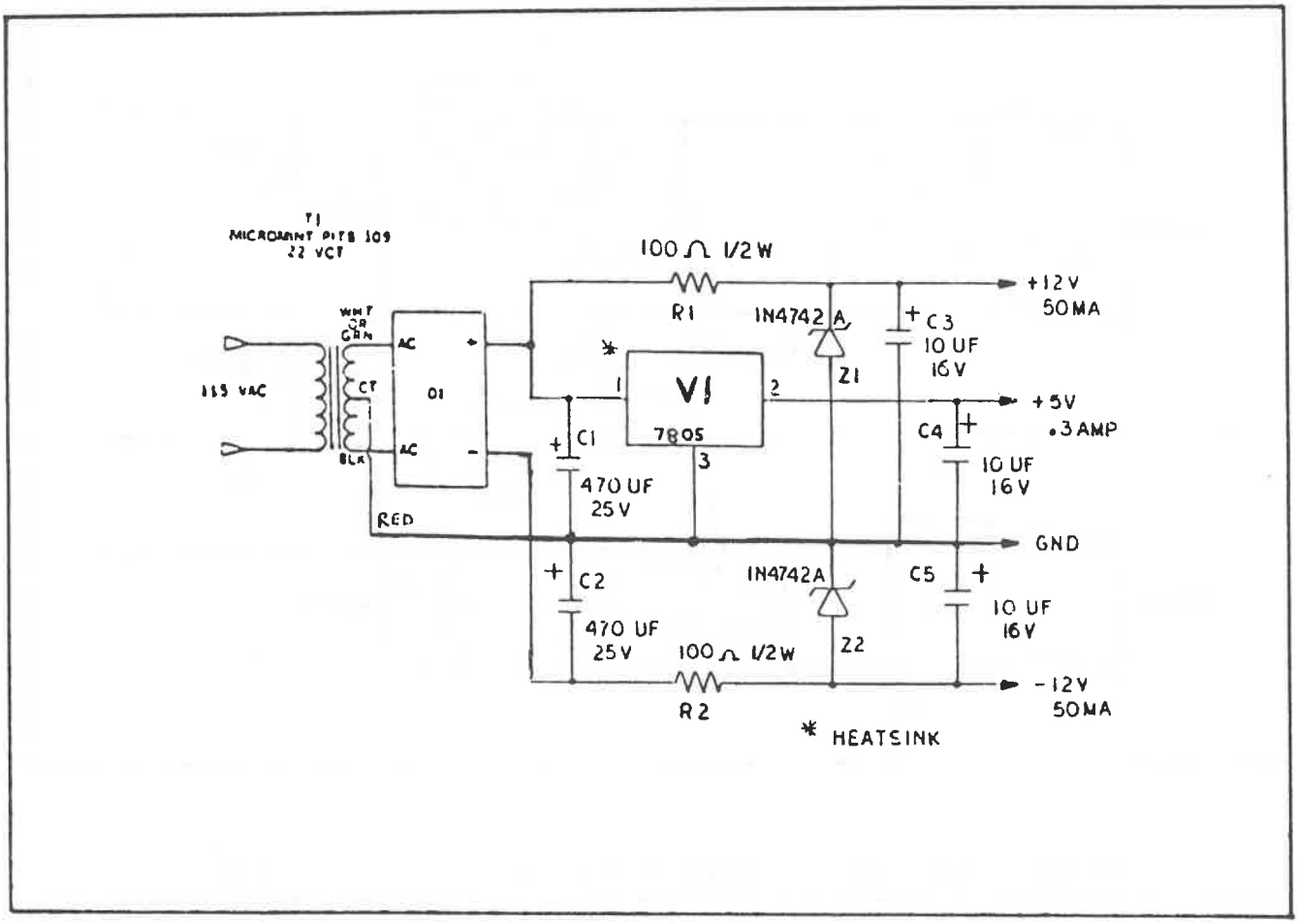
Z8 BASIC SYSTEM/CONTROLLER EXPANSION CONNECTOR J1 (Figure 11)

PIN	SIGNAL	DESCRIPTION
1	+5V	+5 Volt Power Input
2	GND	Signal Ground
3	N/C	No Connection
4	A3/D3	Multiplexed Address/Data Bus
5	A4/D4	Multiplexed Address/Data Bus
6	A5/D5	Multiplexed Address/Data Bus
7	A6/D6	Multiplexed Address/Data Bus
8	A7/D7	Multiplexed Address/Data Bus
9	A0/D0	Multiplexed Address/Data Bus
10	A1/D1	Multiplexed Address/Data Bus
11	A2/D2	Multiplexed Address/Data Bus
12	P2/0	Port 2 bit 0
13	P2/1	Port 2 bit 1
14	P2/2	Port 2 bit 2
15	P2/3	Port 2 bit 3
16	P2/4	Port 2 bit 4
17	P2/5	Port 2 bit 5
18	P2/6	Port 2 bit 6
19	P2/7	Port 2 bit 7
20	R/W	Read/Write Data Strobe
21	DS	Data Strobe
22	AS	Address Strobe
A	-12V	-12 Volt Power Input
B	+12V	+12 Volt Power Input
C	No Connection	
D	No Connection	
E	P3/1	Port 3 bit 1
F	No Connection	
H	P3/2	Port 3 bit 2
J	No Connection	
K	P3/5	Port 3 bit 5
L	P3/6	Port 3 bit 6
M	P3/7	Port 3 bit 7
N	P3/0	Port 3 bit 0
P	P3/4	Port 3 bit 4
R	P3/3	Port 3 bit 3
S	A15	Address Bus
T	A14	Address Bus
U	A13	Address Bus
V	A12	Address Bus
W	A11	Address Bus
X	A10	Address Bus
Y	A9	Address Bus
Z	A8	Address Bus



VIEWED FACING THE CONNECTOR

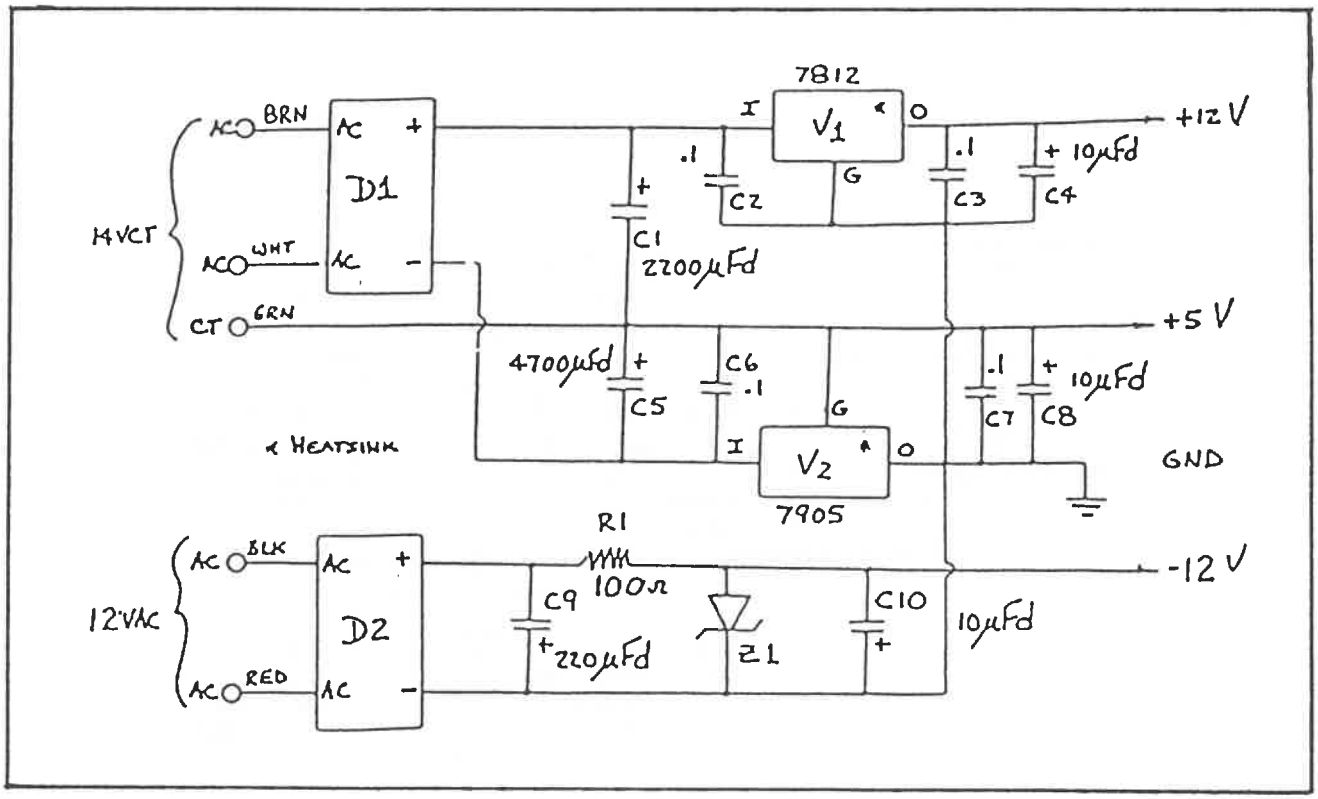
Z8 UNIVERSAL POWER SUPPLY SCHEMATIC (Figure 12)



Z8 UNIVERSAL POWER SUPPLY PARTS LIST (Figure 13)

ITEM	PART # or DESCRIPTION
PCB	U/PWR-PCB Circuit Board
T1	Transformer PITB-109 (22VCT @ 300 Ma.)
V1	MCM7805CT or equiv.
R1,R2	100 Ohm 1/2 Watt (BRN BLK BRN)
Z1,Z2	1N4742A Zener Diode
D1	KBP02 Diode Bridge or equiv.
C1	2200 MFd. 25VDC
C2	470 MFd. 25VDC
C3,C4,C5	10 MFd. 25VDC
Heatsink	THM6071B and THM6072B
Hardware	Screw and nut (3/8" 6-32)
Header	4 pin Molex Connector (with pins)

Z8 STANDARD POWER SUPPLY SCHEMATIC (Figure 14)



Z8 STANDARD POWER SUPPLY PARTS LIST (Figure 15)

ITEM	PART # or DESCRIPTION
PCB	S/PWR-PCB Circuit Board
T1	Transformer PITH-16 (14VCT & 12VAC)
V1	MCM7812CT or equiv.
V2	MCM7905CT or equiv.
R1	100 Ohm 1/2 Watt (BRN BLK BRN)
Z1	1N4742A Zener Diode
D1,D2	KBP02 Diode Bridge or equiv.
C5	4700 MFd. 16VDC
C1	2200 MFd. 25VDC
C9	220 MFd. 25VDC
C4,C8,C10	10 MFd. 25VDC
C2,C3,C6,C7	.1 MFd. 25VDC
Heatsink (V1)	THM6106B-13 or equiv.
Heatsink (V2)	THM6071B and THM6072B
Hardware	2 Each, screw and nut (3/8" 6-32)
Header	4 pin Molex Connector (with pins)

The Z8 BASIC SYSTEM/CONTROLLER is a revision of a circuit originally designed by Steve Ciarcia and presented in "Ciarcia's Circuit Cellar" in BYTE Magazine. The two part article entitled "Build a Z8-Based Control Computer with BASIC", appeared in the July and August 1981 issues.

BASIC/DEBUG is a trademark of ZILOG Inc.

****

WARRANTY

The MICROMINT INC. extends the following warranty.

A factory manufactured circuit board or assembly carries with it a 90 day warranty covering both parts and labor. Any unit which is found to have a defect in materials or workmanship will be repaired or replaced at the option of THE MICROMINT INC. No credit will be given for units which show damage due to user modification or neglect.

Units returned for repair must have prior authorization from THE MICROMINT INC. A return authorization number may be obtained by phone or letter. Please retain a record of the return authorization number as most subsequent correspondence will reference that number. Under no circumstances is any product to be returned to the MICROMINT INC. without prior authorization. THE MICROMINT INC will assume no responsibility for unauthorized returns.

All returns must be shipped prepaid. Insurance is recommended as losses by a shipping carrier are not the responsibilty of THE MICROMINT. Repaired units will be returned with postage paid.

For repair of units which have expired their warranty, a minimum inspection fee must be prepaid. Contact THE MICROMINT for information on current minimum charges

NO WARRANTY is extended on USER ASSEMBLED systems or kits. However, assembled kits will be inspected and repaired with charges based on the current minimum one hour charge. THE MICROMINT INC. retains the right to refuse to repair any USER ASSEMBLED item. This right is at the sole discretion of THE MICROMINT. However, in the event that repair charges would exceed a reasonable amount, the user may be consulted for a determination.

Repairs on user assembled items must be PREPAID.

Return authorization must be obtained prior to any return.

THE MICROMINT reserves the right to change any feature or specification at any time.

****

aiov

aiov

2023.1.11